Protocolli Zero-Knowledge

Crittografia

Luciano Margara

Unibo

2022

Introduzione

Nel 2012 il Turing Award, ovvero il più importante premio scientifico per l'informatica considerato equivalente al Premio Nobel per altre discipline, è stato assegnato a Shafi Goldwasser e Silvio Micali del Massachusetts Institute of Technology. Per la prima volta riceve il premio un italiano

Premio Turing

L'A.M. Turing Award (in italiano, premio Turing) è un premio, assegnato annualmente dalla Association for Computing Machinery (ACM), a una personalità che eccelle per i contributi di natura tecnica offerti alla comunità informatica, in particolare per progressi che siano duraturi e di elevata importanza tecnica.

Premio Turing

Il premio viene spesso anche chiamato "premio Nobel dell'informatica" ed è intitolato al matematico inglese Alan Mathison Turing (1912-1954), in riconoscimento del suo contributo unico e originale alla nascita delle attività di calcolo mediante dispositivi automatici.

Premio Turing

Il fatto che l'ACM si dedichi soprattutto al software e alla teoria dell'elaborazione dei dati ha come effetto che nessuno dei premi riguardi contributi strettamente dedicati all'hardware, settore al quale si dedica invece principalmente l'IEEE.

Premio Turing

Il vincitore è invitato a tenere una lezione in occasione della premiazione, in seguito pubblicata su un periodico scientifico dell'ACM e, a partire dall'anno 2007 riceve un premio in denaro: fino al 2013 ammontava a 250.000 dollari statunitensi ed era offerto da Intel e Google, mentre dall'edizione 2014 la cifra è offerta solo da quest'ultima ed è salita a un milione.

Premio Turing

Il primo vincitore, nel 1966, fu Alan Perlis della Carnegie Mellon University, mentre Frances E. Allen della IBM fu la prima donna a vincere il premio nel 2006, dopo 40 anni dalla sua istituzione.

Premio Turing

Anno	Vincitore	Contributo che ha motivato il riconoscimento	
1966	Alan J. Perlis	*Per la sua influenza nell'area delle tecniche di programmazione avanzata e per la costruzione di compilatori.*[6]	
1967	Maurice V. Wilkes	programma a memorizzazione interna, librerie di programmi	
1968	Richard Hamming	*Per il suo lavoro sui metodi numerici, sistemi di codifica automatica, codici di rilevazione e correzione errori*[7]	
1969	Marvin Minsky	intelligenza artificiale	
1970	James H. Wilkinson	*Per la sua ricerca nell'analisi matematica che ha facilitato l'uso dei moderni computer digitali, in particolare per il lavoro svolto nelle computazioni di algebra lineare e l'analisi degli errori "all'indietro"*[8]	
1971	John McCarthy	*Per il suo libro "The Present State of Research on Artificial Intelligence" che ha dato un'importante contributo nell'area dell'intelligenza artificiale*[9]	
1972	Edsger Dijkstra	Per i suoi numerosi contributi al calcolo automatico e in particolare per l'Algoritmo di Dijkstra[10]	
1973	Charles W. Bachman	*Per i suoi eccezionali contributi alle tecnologie dei database*[11]	
1974	Donald E. Knuth	analisi degli algoritmi e disegno dei linguaggi di programmazione	
1975	Allen Newell Herbert A. Simon	intelligenza artificiale; psicologia della cognizione umana; elaborazione di liste	

Premio Turing

1976	Michael O. Rabin Dana S. Scott	macchine non deterministiche
1977	John Backus	sistemi di programmazione di alto livello; procedure formali per la specificazione dei linguaggi di programmazione
1978	Robert W. Floyd	metodologie per la costruzione di software efficiente e affidabile
1979	Kenneth E. Iverson	linguaggi di programmazione e notazione matematica; implementazione di sistemi interattivi; utilizzi educativi dell'APL; teoria e pratica dei linguaggi di programmazione
1980	C. Antony R. Hoare	definizione e disegno dei linguaggi di programmazione
1981	Edgar F. Codd	sistemi per la gestione delle basi di dati; basi di dati relazionali
1982	Stephen A. Cook	complessità delle computazioni
1983	Ken Thompson Dennis M. Ritchie	teoria generica dei sistemi operativi; implementazione del sistema operativo Unix
1984	Niklaus Wirth	sviluppo dei linguaggi per il computer
1985	Richard M. Karp	teoria degli algoritmi, in particolare teoria della NP-completezza
1986	John Hopcroft Robert Tarjan	disegno e analisi degli algoritmi e delle strutture di dati

Premio Turing

1987	John Cocke	teoria dei compilatori; architettura dei grandi sistemi; sviluppo dei computer I RISC
1988	Ivan Sutherland	computer grafica
1989	William (Velvel) Kahan	analisi numerica
1990	Fernando J. Corbató	CTSS; Multics
1991	Robin Milner	LCF; ML; CCS
1992	Butler W. Lampson	ambienti distribuiti di personal computing
1993	Juris Hartmanis Richard E. Stearns	teoria della complessità computazionale
1994	Edward Feigenbaum Raj Reddy	sistemi di intelligenza artificiale di larga scala
1995	Manuel Blum	teoria della complessità computazionale; sue applicazioni alla crittografia e verifica dei programmi
1996	Amir Pnueli	logica temporale, verifica di programmi e sistemi
1997	Douglas Engelbart	elaborazioni interattive
1998	James Gray	elaborazione delle basi di dati e delle transazioni
1999	Frederick P. Brooks, Jr.	architettura dei computer; sistemi operativi; ingegneria del software

Premio Turing

2000	Andrew Chi-Chih Yao	teoria della computazione incluse generazione di numeri pseudocasuali, crittografia e complessità della comunicazione
2001	Ole-Johan Dahl Kristen Nygaard	programmazione orientata agli oggetti
2002	Ronald L. Rivest Adi Shamir Leonard M. Adleman	crittografia con chiave pubblica
2003	Alan Kay	programmazione orientata agli oggetti
2004	Vinton G. Cerf Robert E. Kahn	internetworking
2005	Peter Naur	Forma di Backus - Naur e *Algol-60*
2006	Frances E. Allen	compilatori di codice
2007	Edmund M. Clarke E. Allen Emerson Joseph Sifakis	Ricerca e sviluppo del model checking
2008	Barbara Liskov	Per aver contribuito allo sviluppo dei linguaggi di programmazione e progettazione di sistemi, specialmente nel campo di astrazione dati, tolleranza degli errori e algoritmi di computazione distribuita

Premio Turing

2009	Charles P. Thacker	Per la progettazione e la realizzazione pionieristica del primo personal computer moderno - lo Xerox Alto - e per le invenzioni ed i contributi alle reti locali (compresa l'Ethernet), le workstation multiprocessore, per gli studi sulla coerenza dei protocolli cache e per i Tablet PC
2010	Leslie G. Valiant	*Per i contributi alla teoria della computabilità, con la teoria nota come Probably Approximately Correct learning (PAC learning), alla teoria della complessità computazionale, con la teoria della complessità dell'enumerazione, alla teoria del calcolo algebrico ed alla teoria del calcolo parallelo e distribuito.*[12]
2011	Judea Pearl	*Per i contributi fondamentali all'intelligenza artificiale attraverso lo sviluppo di un calcolo di ragionamento probabilistico e causale.*[13]
2012	Silvio Micali Shafi Goldwasser	Per i contributi che hanno permesso di gettare le basi della teoria della complessità nel campo della crittografia e per aver sperimentato nuovi metodi per la verifica efficiente di dimostrazioni matematiche nel campo della teoria della complessità.[14]
2013	Leslie Lamport	Per i contributi pionieristici nello studio dell'affidabilità e della consistenza dei sistemi di calcolo: "per i suoi contributi fondamentali alla teoria e alla pratica dei sistemi distribuiti e concorrenti, in particolare per l'invenzione di concetti quali causalità, orologi logici, *safety* e *liveness*, le *replicated state machines*, e la consistenza sequenziale[15]
2014	Michael Stonebraker	Per i suoi fondamentali contributi ai concetti e alla pratica dei moderni sistemi di basi di dati.[16]

Premio Turing

2015	Whitfield Diffie Martin Hellman	Per i contributi fondamentali alla crittografia moderna. L'opera "*New Directions in Cryptography*" pubblicata nel 1976 da Diffie e Hellman ha introdotto le idee di crittografia a chiave pubblica e le firme digitali, alla base dei protocolli di sicurezza di maggior uso su Internet in quel periodo.[17]
2016	Tim Berners-Lee	Per l'invenzione del World Wide Web, del primo browser web e dei protocolli e degli algoritmi fondamentali che consentono al web di evolvere.[18]
2017	John L. Hennessy David A. Patterson	Per un pionieristico approccio sistematico e quantitativo alla progettazione e alla valutazione di architetture di computer con un impatto duraturo nel settore dei microprocessori.[19]
2018	Yoshua Bengio Geoffrey Hinton Yann LeCun	Per scoperte concettuali e ingegneristiche che hanno reso le reti neurali profonde un componente critico dell'informatica.
2019	Edwin Catmull Patrick M. Hanrhan	Per i contributi fondamentali alla Computer grafica 3D, e l'impatto della Computer-generated imagery (CGI) nel cinema ed altre applicazioni.
2020	Alfred Aho Jeffrey Ullman	Per gli algoritmi fondamentali e la teoria sottostante l'implementazione dei linguaggi di programmazione e per aver raccolto questi risultati e quelli di molti altri all'interno di libri altamente influenti che hanno contribuito all'educazione di generazioni di informatici.[20]
2021	Jack Dongarra	Per il pionieristico contributo agli algoritmi di calcolo numerico e librerie che hanno permesso al software dedicato al calcolo ad alte prestazioni di tenere il passo con l'evoluzione esponenziale dell'hardware negli ultimi 40 anni.[21]
2022	Robert Metcalfe	Per l'invenzione, standardizzazione e commercializzazione di Ethernet.[22]
2023	Avi Wigderson	Per la ridefinizione del ruolo della casualità nella computazione e il suo decennale ruolo di leadership intellettuale nell'informatica teorica.[23]

Premio Turing

Nazione	Numero premi
Stati Uniti	49
Regno Unito	7
Israele	5
Canada	3
Norvegia	2
Danimarca	1
Grecia	1
India	1
Italia	1
Paesi Bassi	1
Svizzera	1
Venezuela	1
Totale	73

Introduzione

I loro studi hanno illustrato l’importanza degli eventi casuali nella crittografia dando poi luogo ai metodi di "dimostrazione a conoscenza zero" e costituiscono una pietra miliare nell’informatica teorica

Introduzione

In un protocollo Zero-Knowledge ci sono ~~sue~~ due attori. Un prover P (dimostratore) e un verifier V (verificatore). Il prover P deve dimostrare al verifier V di essere in possesso di una facoltà particolare o di una conoscenza specifica su un argomento senza comunicargli alcun'altra informazione salvo l'evidenza del suo possesso di tale facoltà o conoscenza

Protocolli Zero-Knowledge: un esempio

P afferma che, condotto su una qualsiasi spiaggia, è in grado con un'occhiata di contarne i granelli di sabbia

V controllerà che questa affermazione è vera con una probabilità arbitrariamente prossima a 1 (certezza) da lui stesso stabilita, senza però ricavare alcun'altra informazione sul metodo seguito da P per contare i granelli

Protocolli Zero-Knowledge: un esempio

P e V adottano un protocollo iterativo consistente in un numero $k+1$ di domande e risposte ove il valore di k è scelto da V. Per semplificare ulteriormente le cose ammettiamo che V si accontenti che le risposte di P siano costituite da un bit che indica se il numero di granelli è pari o dispari. Alla iterazione $i = 0, 1, 2, ..., k$ la risposta deve essere $b_i = 0$ se il numero di granelli è pari, $b_i = 1$ se è dispari

Protocolli Zero-Knowledge: un esempio

Si noti che V non è in grado di contare i granelli di sabbia ma vuole comunque verificare se P è in grado di farlo.
V non può chiedere a P come fa a contare i granelli di sabbia

Iterazione i-esima del protocollo

- V chiede a P di voltarsi per non osservare quello che farà
- V sceglie un bit random e lanciando una moneta
- Se $e = 0$, V toglie un granello di sabbia e lo intasca, altrimenti non fa nulla
- V chiede a P di guardare di nuovo la spiaggia e di calcolare il nuovo valore b_i
- Se $(e = 0\ AND\ b_i \neq b_{i-1})$ oppure $(e = 1\ AND\ b_i = b_{i-1})$, V passa all'iterazione $i + 1$, altrimenti arresta il procedimento dichiarando che P è un impostore

Protocolli Zero-Knowledge: un esempio

1 ▷ P sa contare i granelli di sabbia. V, seguendo il protocollo, otterrà k risposte esatte da P e quindi sarà in gradi di verificare che P sa come contare i granelli con probabilità 1

2 ▷ P non sa contare i granelli di sabbia. Ad ogni iterazione del protocollo P risponderà in modo errato con probabilità $1/2$ e in modo corretto con probabilità $1/2$. La probabilità complessiva che P possa ingannare V è $\frac{1}{2^k}$

Protocolli Zero-Knowledge: proprietà

- **Completezza**: se l'affermazione di P è vera, V ne accetta sempre la dimostrazione.
  (cioè P sa effettivamente contare)
- **Correttezza**: se l'affermazione di P è falsa (P è disonesto), V può essere convinto della veridicità di tale affermazione solo con una probabilità $\leq \frac{1}{2^k}$ per un valore arbitrario k scelto da lui stesso
  (cioè P non sa contare, ma finge di saperlo fare)
  (cioé P riesce ad ingannare V con probabilitá di 1/2^k)
- **Conoscenza-Zero**: Se l'affermazione di P è vera nessun verificatore V, anche se disonesto (nell'uso del protocollo), può acquisire alcuna informazione su questo fatto salvo la sua veridicità
  (cioè V non é in grado di ottenere maggiori informazioni su come P riesca a contare)

Protocolli Zero-Knowledge: identificazione

Un prover P deve dimostrare la propria identità a un verifier V e non deve svelare altro che la sua identità né a V né a possibili intrusi.
Di seguito presentiamo il primo protocollo di identificazione zero-knowledge proposto da Fiat e Shamir, che costituisce uno schema generale per tutti i protocolli successivi.

Di seguito, ci sono 3 esempi senza utilizzo di Protocolli Zero-Knowledge

Documento identità 1

id: nome e cognome

Sulla linea: nome e cognome, fotografia

faccia

Privato (cioè non mandato sulla linea di comunicazione)

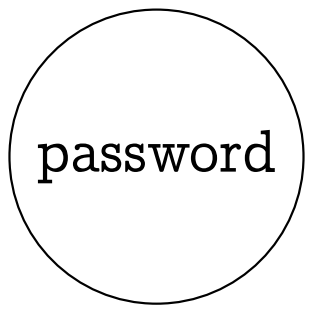
Fatto: esiste un essere umano "Sergio Mattarella" con la faccia

Identificazione: ti dimostro che ho la faccia di "Sergio Mattarella".

Login e Password 2

id: login

Sulla linea: login, password



Privato (cioé non deve passare sulla linea di comunicazione)

Fatto: esiste un essere umano con login "sergio.mattarella" che ha scelto la password "vivalitalia41"

Identificazione: ti dimostro che conosco la password "vivalitalia41".

Cifrari Asimmetrici 3

id: chiave pubblica

Sulla linea: chiave pubblica, challenge firmato con chiave privata

Privato (cioé non deve passare sulla linea di comunicazione)

Fatto: esiste una chiave pubblica "K_{pub}" associata a una chiave privata "K_{priv}"

Identificazione: ti dimostro che conosco la chiave privata "K_{priv}".

Protocolli Zero-Knowledge: identificazione

L'identificazione può essere ottenuta mediante una password su un canale sicuro, o mediante un cifrario asimmetrico su un canale insicuro. Nel secondo caso, il metodo può essere messo in pericolo dal fatto che V chieda a P di applicare la sua chiave privata a una sequenza r che V stesso ha generato. Un verifier disonesto potrebbe generare una r ad arte per carpire qualche informazione sulla chiave privata di P

Protocolli Zero-Knowledge: identificazione

Consideriamo dunque un protocollo zero-knowledge con cui P dimostra la sua identità senza svelare alcun'altra informazione basato sulla difficoltà di invertire la funzione potenza nell'algebra modulare, ovvero di calcolare la radice in modulo

Radice quadrata modulare

Sia n un numero non primo

$$t = s^2 \ mod \ n$$

Calcolare t, dati s ed n, è facile. Calcolare s, (radice quadrata di t) dati t ed n, è esponenzialmente difficile se non si conosce la fattorizzazione di n

Protocollo di identificazione di Fiat-Shamir

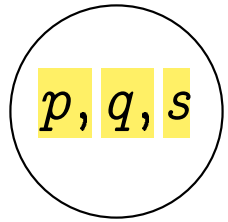
P sceglie $n = pq$ con p, q primi, e un intero positivo $s < n$.
Calcola $t = s^2 \ mod \ n$. Rende nota la coppia $\langle n, t \rangle$ come una sorta di chiave pubblica: il suo segreto è la terna $\langle p, q, s \rangle$ e nessuno di questi valori può essere desunto in tempo polinomiale da $\langle n, t \rangle$. Quindi può partire il protocollo i cui passi, iterati per k volte, consentono a V di stabilire l'identità di P con probabilità $1 - \frac{1}{2^k}$ senza nulla poter inferire sul segreto di P

Protocollo di identificazione di Fiat-Shamir

< n,t > noti, senza far passare altro sulla linea di comunicazione: < p,q,s > privati

id: n, t

Sulla linea: ...



Privato (segreto)

Protocollo di identificazione di Fiat-Shamir

1. V chiede a P di iniziare una iterazione
2. P genera un intero random $r < n$, calcola $u = r^2 \ mod \ n$ e comunica u a V
3. V genera un intero random e in $\{0, 1\}$ e comunica e a P
4. P calcola $z = rs^e \ mod \ n$ e comunica z a V
   se $e = 0$ si ha $z = r$, se $e = 1$ si ha $z = rs \ mod \ n$
5. V calcola $x = z^2 \ mod \ n$, se $(x = ut^e \ mod \ n)$ ripete dal passo 1 altrimenti blocca il protocollo senza identificare P

Protocollo di Fiat-Shamir: Completezza

Completezza.

Se P è onesto la condizione $x = ut^e \ mod \ n$ del passo 5 è sempre verificata

Se $e = 0$ si ha $x = r^2 \ mod \ n$ e $ut^e \ mod \ n = r^2 \ mod \ n$

Se $e = 1$ si ha $x = r^2 s^2 \ mod \ n$ e

$ut^e \ mod \ n = r^2 s^2 \ mod \ n$

Dunque V accetta l'asserzione di P per qualunque valore di k

Protocollo di Fiat-Shamir

1. V chiede a P di iniziare una iterazione
2. V genera un intero random e in $\{0, 1\}$ e comunica e a P
3. P genera un intero random $r < n$ e comunica $u = r^2 \ mod \ n$ a V se $e = 0$, altrimenti gli comunica $u = r^2/t \ mod \ n$
4. P comunica $z = r$ a V
5. V calcola $x = z^2 \ mod \ n = r^2 \ mod \ n$, che sarà sempre uguale a $ut^e \ mod \ n$ e quindi ripete dal passo 1, sempre.

Protocollo di Fiat-Shamir: Correttezza

Correttezza. Se P è disonesto (cioè prova a farsi identificare come un altro utente) e conosce il valore di t che è pubblico ma non conosce il valore di s, può tentare di ingannare V prevedendo i valori di e che questi genererà al passo 3 del protocollo. Così, al passo 2, P invierà a V il valore $u = r^2 \ mod \ n$ se prevede di ricevere $e = 0$, o il valore $u = r^2/t \ mod \ n$ se prevede di ricevere $e = 1$. In entrambi i casi invierà poi a V lo stesso valore $z = r$ al passo 4

Protocollo di identificazione di Fiat-Shamir

Se la previsione di P su e è corretta, al passo 5, per entrambi i valori di e, V scoprirà che $x = ut^e \ mod \ n = r^2 \ mod \ n$ e accetterà l'iterazione; ma se la previsione è sbagliata la relazione non sarà verificata e sarà scoperto l'inganno. La proprietà di correttezza segue dall'osservazione che, poiché e viene generato a caso, le previsioni di P su e sono corrette con probabilità $1/2$

Fiat - Shamir